日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　３月１４日

出願番号
Application Number: 特願２００１－０７１６８０

出願人
Applicant(s): 株式会社東芝

２００１年　７月２７日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号　出証特２００１－３０６５０１３

【書類名】　特許願
【整理番号】　1FB00Y0051
【あて先】　特許庁長官殿
【国際特許分類】　G06F 13/00
【発明の名称】　クラスタシステム
【請求項の数】　4
【発明者】
【住所又は居所】　東京都府中市東芝町１番地　株式会社東芝　府中事業所内
【氏名】　平山　秀昭
【特許出願人】
【識別番号】　000003078
【氏名又は名称】　株式会社　東芝
【代理人】
【識別番号】　100083161
【弁理士】
【氏名又は名称】　外川　英明
【電話番号】　(03)3457-2512
【手数料の表示】
【予納台帳番号】　010261
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　クラスタシステム

【特許請求の範囲】

【請求項１】

複数の電子計算機と、共有ディスク装置と、前記複数の電子計算機上で動作するプロセスから前記共有ディスク装置に記録されたファイルに対して共有アクセスするためにロック機能によりデータの一貫性を保持するための排他制御をするクラスタ共有ファイルシステムとを持ったクラスタシステムであって、

前記クラスタ共有ファイルシステムが管理するファイルを前記プロセスのアドレス空間内にマッピングして共有メモリを設定するためのクラスタ共有メモリ設定手段と、

前記クラスタ共有ファイルシステムのロック機能を前記共有メモリにアロケーションして前記共有メモリ上でデータの一貫性を保持するための排他制御を可能とするクラスタ共有メモリ用ロックアロケーション手段とを備えたことを特徴とするクラスタシステム。

【請求項２】

前記設定された共有メモリ内の全てのページに対するアクセスを禁止にするアクセス禁止設定手段と

このアクセス禁止が設定されたページをアクセスしたときにＲＥＡＤページフォールトが発生した場合、そのページに前記マッピングされたデータを前記共有ディスク装置に記録されたファイルから読み出して書き込むデータ書き込み手段と、

このデータが書き込まれたページを読み出し可能に設定する設定手段とを具備することを特徴とする請求項１記載のクラスタシステム。

【請求項３】

前記設定された共有メモリ内の全てのページに対するアクセスを禁止にするアクセス禁止設定手段と

このアクセス禁止が設定されたページをアクセスしたときにＷＲＩＴＥページフォールトが発生した場合、そのページに前記マッピングされたデータを前記共

有ディスク装置に記録されたファイルから読み出して書き込むデータ書き込み手段と、

このデータが書き込まれたページを読み出し／書き込み可能に設定する設定手段とを具備することを特徴とする請求項１記載のクラスタシステム。

【請求項４】

前記クラスタ共有メモリをロック操作した場合、前記クラスタ共有ファイルシステムが管理するファイルを前記プロセスのアドレス空間にマッピングしたクラスタ共有メモリ領域のページをアクセス不許可に設定してロックを取得する手段と、

この取得したロックをアンロック操作した場合、前記クラスタ共有メモリ領域の更新されたページのデータを前記クラスタ共有ファイルシステムが管理するファイルに書き戻す手段を具備したことを特徴とする請求項３記載のクラスタシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、記憶装置と計算機を専用の高速ネットワークで接続したＳＡＮ（Ｓｔｏｒａｇｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（ストレージエリア・ネットワーク））環境における並列処理プログラミングが容易なクラスタシステムに関するものである。

【０００２】

【従来の技術】

近年、データのストレージ装置としての磁気ディスク装置と電子計算機（以下、サーバーコンピュータまたは単にサーバーと呼ぶ）とをＦｉｂｅｒ　Ｃｈａｎｎｅｌ（ファイバ・チャネル）等の専用の高速ネットワークで接続したＳＡＮが注目を浴びている。

【０００３】

このＳＡＮには以下のような利点がある。（１）複数のサーバーがストレージ装置を共有することができる。（２）ストレージ装置のアクセス負荷をＬＡＮ（

Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）から分離させることができる。（３）ファイバ・チャネル等によりストレージ装置へのアクセスを高速にすることができる。このうち（１）はＳＡＮの一般的な利点であるが、ここにクラスタ共有ファイルシステムの技術を用いると、単に複数のサーバーがストレージ装置を共有できるだけでなく、ファイルを共有アクセスすることが可能になる。

【０００４】

米国Ｓｉｓｔｉｎａ　Ｓｏｆｔｗａｒｅ　Ｉｎｃが開発しているＧＦＳ（Ｇｌｏｂａｌ　Ｆｉｌｅ　Ｓｙｓｔｅｍ）は、このようなクラスタ共有ファイルシステムの一例である。クラスタ共有ファイルシステムでは、複数のサーバーコンピュータによるストレージ装置に記憶されたファイルへの共有アクセスを可能にしている。

【０００５】

複数のサーバーコンピュータによるファイルへの共有アクセスというと、一般的にはＮＦＳ（Ｎｅｔｗｏｒｋ　Ｆｉｌｅ　Ｓｙｓｔｅｍ）が連想されるが、ＮＦＳでは複数のサーバーコンピュータで同一ファイルを更新した場合のデータの一貫性を保証していない。これに対しクラスタ共有ファイルシステムではデータの一貫性を保証している。

【０００６】

クラスタ共有ファイルシステムは、複数のサーバーコンピュータによるファイルの共有アクセス（ＲＥＡＤ／ＷＲＩＴＥ）機能の他に、データの一貫性を保証するために複数のサーバーコンピュータに跨るロック機構であるクラスタ共有ファイルシステム用ロック機構を持っている。このクラスタ共有ファイルシステム用ロック機構により、複数のサーバーコンピュータから構成されるクラスタシステム（Ｃｌｕｓｔｅｒ　Ｓｙｓｔｅｍ）上で並列アプリケーションプログラムが実行可能となる。

【０００７】

例えば、クラスタ共有ファイルシステムとクラスタ共有ファイルシステム用ロック機構とを実装した複数のサーバーコンピュータがＳＡＮを介して１つの磁気ディスク装置に接続されているクラスタシステムについて考察する。このクラス

タシステムは、主メモリを共有していない疎結合のクラスタシステムである。各サーバーコンピュータ上で実行されているプロセスは、クラスタ共有ファイルシステムを用いることにより、磁気ディスク装置に記憶されているファイルに共有アクセスすることができる。しかも、プロセスは、クラスタ共有ファイルシステム用ロック機構を用いて排他制御処理をすることにより、磁気ディスク装置に記憶されているファイルのデータの一貫性を保証することができる。

【０００８】

これに対して、ファイルシステムと共有メモリシステムとロック機構とが実装され、複数のプロセッサを持ち、例えば１つの磁気ディスク装置が接続されているＳＭＰ型（Ｓｙｍｍｅｔｒｉｃａｌ　Ｍｕｌｔｉｐｒｏｃｅｓｓｏｒ）並列計算機について考察する。このＳＭＰ型並列計算機上で実行されている複数のプロセスは、それぞれファイルシステムを介して磁気ディスク装置に記憶されたファイルに共有アクセスしたり、共有メモリシステムを介して共有メモリ（主メモリ）にアクセスすることができる。また、複数のプロセスは、ロック機構を介して磁気ディスク装置に記憶されたファイルや共有メモリに記憶されたデータに対する共有アクセスにおける排他制御処理をすることにより、データの一貫性を保持することができる。

【０００９】

このように従来の疎結合のクラスタトステムとＳＭＰ型並列計算機とを比較すると、両者ともに磁気ディスク装置に記憶されたファイルに対する共有アクセスおよびこれらに対するデータの一貫性の保持は可能である。しかし、クラスタシステムでは、複数のサーバーコンピュータから共有メモリへの共有アクセスをすることができない。

【００１０】

【発明が解決しようとする課題】

クラスタ共有ファイルシステムを持つクラスタシステムでは、ファイルへの共有アクセスはできるが、メモリへの共有アクセスはできない。このため、クラスタ共有ファイルシステムを持つクラスタシステムでは、ＳＭＰ型並列計算機と比べると、その実行するアプリケーションプログラムを並列プログラムにて記述す

るのが困難であるという課題がある。具体的には、複数のサーバーコンピュータ上で実行されているプロセスでファイル（又はデータ）を共有して互いにデータの一貫性を確保しながらデータの同期をとってデータ処理するプログラミングを記述する場合には、プロセスが共有して処理するファイル（又はデータ）をメモリ上に配置して処理することが必要であるが、これをすることができない。もし、プロセスがファイル（又はデータ）を同期をとりながらデータの一貫性を確保して共有処理することが必要な場合には、クラスタ共有ファイルシステムを利用して、その共有対象のファイル（又はデータ）を磁気ディスク装置上に配置して処理することが必要である。この場合には、プロセスが、ファイル（又はデータ）に対する処理を入出力装置を対象とするＩ／Ｏ処理のコマンドを使用してプログラミングされている必要がある。このようにファイル装置（磁気ディスク装置）に記憶されているファイル（又はデータ）のデータ処理する場合は、メモリ上に配置されたファイル（又はデータ）を処理する場合に比べて、プログラムの記述が複雑になる。これは、メモリ上に配置されたデータに対する処理であれば、ｌｏａｄ命令やｓｔｏｒｅ命令を使用して簡単なプログラム記述で処理が記述できる。

【００１１】

しかし、これに対して磁気ディスク装置上に配置されたファイルやデータを処理する場合には、Ｉ／Ｏ処理をするための複雑なコマンドを使用したプログラムを記述する必要があり、並列プログラムにて記述するのが困難である。また、メモリ上に配置されたデータに対するデータ処理に比べて、磁気ディスク装置などのファイル装置（Ｉ／Ｏ装置）上に配置されたデータを処理する場合には、その処理時間が多くかかり、処理速度が落ちるという問題もある。

【００１２】

以上の説明したように、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムでは、磁気ディスク装置などのファイル装置上に記憶されたファイルへの共有アクセスは可能であるが、主メモリへの共有アクセスは可能でないため、ＳＭＰ型並列計算機と比べると、並列プログラムにて記述するのが困難であるという課題がある。

【００１３】

本発明は、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムにおいて、並列プログラムの記述が容易なクラスタシステムを提供することを目的とする。

【００１４】

【課題を解決するための手段】

本発明は、複数の電子計算機と、共有ディスク装置と、前記複数の電子計算機上で動作するプロセスから前記共有ディスク装置に記録されたファイルに対して共有アクセスするためにロック機能によりデータの一貫性を保持するための排他制御をするクラスタ共有ファイルシステムとを持ったクラスタシステムであって、前記クラスタ共有ファイルシステムが管理するファイルを前記プロセスのアドレス空間内にマッピングして共有メモリを設定するためのクラスタ共有メモリ設定手段と、前記クラスタ共有ファイルシステムのロック機能を前記共有メモリにアロケーションして前記共有メモリ上でデータの一貫性を保持するための排他制御を可能とするクラスタ共有メモリ用ロックアロケーション手段とを備えたことを特徴とする。

【００１５】

本発明によれば、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムにおいて、並列プログラムの記述を容易にすることができる。

【００１６】

【発明の実施の形態】

本発明のポイントは、クラスタシステムにおいて、アプリケーションプログラムを実行することで生成されるプロセスが共有ディスク装置に記録されたファイルを、クラスタ共有ファイルシステムを用いてプロセスが主メモリ上に配置されているアドレス空間内に仮想的に設けたクラスタ共有メモリ（分散共有メモリ）領域上にマッピングすることで、ファイルへのアクセスを主メモリへのアクセスとして処理することである。

【００１７】

以下、図面を用いて、本発明の一実施形態を詳細に説明する。図１は、本発明のクラスタシステムを示す図である。図１において、クラスタシステム１０は、ファイバ・チャネルで構成されているストレージエリア・ネットワーク（ＳＡＮ）１５にサーバーコンピュータ１１、サーバーコンピュータ１２、サーバーコンピュータ１３、サーバーコンピュータ１４と磁気ディスク装置で構成される共有ディスク装置１６が接続されて構成されている。

【００１８】

各サーバーコンピュータ１１～１４は、図２に示すように構成されている。図２では、サーバーコンピュータ１１～１４の代表として、サーバーコンピュータ１１の構成を説明する。図２において、サーバーコンピュータ１１には、主メモリ１１１、クラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル１１３、クラスタ共有メモリ用ロック操作手段１１４、クラスタ共有メモリ用ページ操作手段１１５、クラスタ共有ファイルシステム用ロック機構１１６、クラスタ共有ファイルシステム１１７、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８、更新ページリスト１１９、クラスタ共有メモリマップ手段１２０、クラスタ共有メモリアンマップ手段１２１、クラスタ共有メモリ用ロックアロケーション手段１２２、クラスタ共有メモリ用ロック解放手段１２３が設けられて構成されている。サーバーコンピュータ１１では、あるアプリケーションプログラムが実行されることで生成されるプロセス１１２が主メモリ１１１上のアドレス空間内に配置されて図示しないプロセッサにより実行されて動作している。

【００１９】

図３は、プロセス１１２の詳細を示す図である。図３において参照符号Ａは、プロセス１１２のアドレス空間内におけるアドレスを示す。プロセス１１２には、アドレスＡを先頭として複数のページＰ０、Ｐ１、Ｐ２、Ｐ３、Ｐ４、Ｐ５が設けられてる。このプロセス１１２内に仮想的にクラスタ共有メモリが設けられる。

【００２０】

クラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テ

ーブル１１３は、図４に示すようにクラスタ共有メモリ用ロックＩＤ欄１１３ａとクラスタ共有ファイルシステム用ロックＩＤ欄１１３ｂとから構成されている。このクラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル１１３にクラスタ共有メモリ用ロックＩＤとクラスタ共有ファイルシステム用ロックＩＤとを登録することにより、両者の対応を関係づける。

【００２１】

クラスタ共有メモリ用ロック操作手段１１４は、クラスタ共有メモリ内に設けられる各ページに記憶されたデータの一貫性を確保するための排他制御をするためにロックを設定（ロックする）し、また設定したロックを解除（アンロック）するための操作手段である。クラスタ共有メモリ用ページ操作手段１１５は、クラスタ共有メモリに設けられ各ページに対するデータの記憶、各ページに記憶されたデータの読み出しを行う操作手段である。

【００２２】

クラスタ共有ファイルシステム１１７は、共有ディスク装置１６に記録されたファイルをサーバコンピュータ１１、サーバコンピュータ１２、サーバコンピュータ１３、サーバコンピュータ１４とで共有するためのファイルシステムである。クラスタ共有ファイルシステム用ロック機構１１６は、共有ディスク装置１６に記録されたファイルをサーバコンピュータ１１、サーバコンピュータ１２、サーバコンピュータ１３、サーバコンピュータ１４とで共有する際に、データの一貫性を確保するための排他制御をするためにロックを設定（ロックする）し、また設定したロックを解除（アンロック）するための操作を行う機構である。

【００２３】

クラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８は、図５に示すように、アドレス欄１１８ａ、サイズ欄１１８ｂ、ファイル名欄１１８ｃ、ファイル記述子１１８ｄ、オフセット欄１１８ｅとから構成されている。このクラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８は、共有ディスク装置１６に記録されたファイルをクラスタ共有メモリに対応付け（マッピング）するために設けられている。クラスタ共有メモリマップ手段１２０は、プロセス１１２のアドレス空間内にクラスタ共有メモリを仮想的に設けるための手段である

【００２４】

クラスタ共有メモリアンマップ手段１２１は、プロセス１１２のアドレス空間内に仮想的に設けられたクラスタ共有メモリを解放するための手段である。

クラスタ共有メモリ用ロックアロケーション手段１２２は、クラスタ共有メモリ用ロックと一対一に対応するクラスタ共有ファイルシステム用ロックとをアロケーションするための手段である。クラスタ共有メモリ用ロック解放手段１２３は、クラスタ共有メモリ用ロックアロケーション手段１２２により、アロケーションされたクラスタ共有メモリ用ロックとクラスタ共有ファイルシステム用ロックとを解放するための手段である。

【００２５】

図６は、更新ページリスト１１９の詳細を示す図である。この更新ページリスト１１９は、共有メモリの中のデータが更新されたページのページ番号を記録するためのリストである。

【００２６】

以下、図７を用いてサーバーコンピュータ１１で実行されているプロセス１１２が共有ディスク装置１６に記録されているファイルに対してアクセスする場合の動作を詳細に説明する。

【００２７】

図７は、クラスタ共有メモリを用いたクラスタ共有ファイルへのアクセス動作の処理手順を示すフローチャート図である。まず、プロセス１１２が共有ディスク装置１６に記録されている共有ファイルに対してアクセスしようとする場合には、プロセス１１２のアドレス空間内に仮想的に共有メモリを設け、この共有メモリ上にアクセス対象の共有ファイルをマッピングする（ステップＳ７０）。具体的には、プロセス１１２がクラスタ共有メモリマップ手段１２０にクラスタ共有メモリマップ操作を指示する。このプロセス１１２からの指示に基づいてクラスタ共有メモリマップ手段１２０は、プロセス１２０が配置された主メモリ１１１上のアドレス空間内に共有メモリをマッピングする。

【００２８】

このクラスタ共有メモリマップ手段１２０による共有メモリのマッピング操作手順を図８に示すフローチャートを用いて説明する。まず、ステップＳ８０で、プロセス１１２のアドレス空間内にクラスタ共有メモリのための領域をアロケーションし、その領域内の全てのページをアクセス不許可に設定する。アロケーションは、プロセス１１２にＣ言語の関数ｍａｌｌｏｃ（）を記述することで実施できる。

【００２９】

次に、ステップＳ８１で、アクセス対象の共有ファイルのプロセス１１２から指定されたオフセット位置から指定されたサイズ分のデータをステップＳ８０でアロケーションした領域にマッピングしたことをクラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８に登録する。この登録の結果として、図５に示すようにプロセス１１２のアドレスＡからサイズＬの範囲にファイル名が「ＤＤＤＤ」で、ファイル記述子が「７」で、オフセット「０」を登録する。ここまでの処理で、プロセス１１２が共有ファイルへのアクセスを共有メモリへのアクセスとして処理するための環境を整えたことになる。

【００３０】

続いてステップＳ７１において、プロセス１１２が共有ファイルへのアクセスを共有メモリへのアクセスとして処理するのに必要な排他制御処理をするためにロックを取得する。そこでロック取得の前準備として、プロセス１１２は、クラスタ共有メモリ用ロックとクラスタ共有ファイル用ロックとを対応づけるために、クラスタ共有メモリ用ロックアロケーション手段１２２にクラスタ共有メモリ用ロックアロケーション操作を指示する。

【００３１】

図１０にクラスタ共有メモリ用ロックアロケーション手段１２２によるクラスタ共有メモリ用ロックアロケーション操作手順のフローチャートを示す。ステップＳ１００において、クラスタ共有メモリ用ロックと一対一に対応するクラスタ共有ファイルシステム用ロックをアロケーションするために、図４に示すクラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル１１３にクラスタ共有メモリ用ロックのＩＤ番号とクラスタ共有ファイルシステム

用ロックのＩＤ番号とを対応づけて登録する。

【００３２】

次にプロセス１１２は、ロックを取得するために、クラスタ共有メモリ用ロック操作手段１１４にロックの取得を指示する。このロック取得の指示を受けたクラスタ共有メモリ用ロック操作手段１１４のロック操作の処理手順を図１２に示したフローチャートを用いて説明する。

【００３３】

まずステップＳ１２０において、クラスタ共有メモリ上の全てのページをアクセス不許可に設定する。次にステップＳ１２１において、図４に示したクラスタ共有メモリ用ロック/クラスタファイルシステム用ロック変換テーブル１１３を用いて、ロック操作されるクラスタ共有メモリ用ロックをクラスタ共有ファイルシステム用ロックに変換する。続いて、ステップＳ１２２において、クラスタ共有メモリ用ロック操作手段１１４は、クラスタ共有ファイルシステム用ロック機構１１６に指示して、変換したクラスタファイルシステム用ロックをロック操作してロックを取得する。

【００３４】

ステップＳ７１でロック取得が成功した場合には、ステップＳ７２に進む。もし、ロック取得が失敗した場合には、換言すると既に他のプロセスが同様な処理により、実体としての共有ファイルのロックを取得している場合には、ロックが取得できないので、ロックが解放されて、取得できるまで処理を待機しステップＳ７１の処理を再び続ける

【００３５】

続いてステップＳ７２において、プロセス１１２は、クラスタ共有メモリ用ページ操作手段１１５に指示して、共有メモリへのＲＥＡＤアクセス（共有メモリに対するｌｏａｄ命令の実行）又はＷＲＩＴＥアクセス（共有メモリに対するｓｔｏｒｅ命令の実行）を実行させる。このクラスタ共有メモリ用ページ操作手段１１５が共有メモリへアクセスを実行すると、この共有メモリの全てのページはアクセス不許可に設定されているため、ページフォールトが発生する。クラスタ共有メモリ用ページ操作手段１１５がＲＥＡＤアクセスした場合には、ＲＥＡＤ

ページフォールトが発生する。また、クラスタ共有メモリ用ページ操作手段１１５がＷＲＩＴＥアクセスした場合には、ＷＲＩＴＥページフォールトが発生する。

【００３６】

ＲＥＡＤページフォールトが発生が発生した場合のクラスタ共有メモリ用ページ操作手段１１５の処理手順を図１４に示す。ＷＲＩＴＥページフォールトが発生した場合のクラスタ共有メモリ用ページ操作手段１１５処理手順を図１５に示す。

【００３７】

図１４を参照して、ＲＥＡＤページフォールトが発生した場合のクラスタ共有メモリ用ページ操作手段１１５の処理手順を説明する。まず、ステップＳ１４０で、ページフォールトが発生した共有メモリ内のページのデータをクラスタ共有ファイルシステム１１７によりクラスタ共有ファイル（共有ディスク装置１６）の対応する部分から当該共有メモリ内のページに読み込む。この時、ページのデータをクラスタ共有ファイルのどの部分から読み込むかは、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８に登録されたデータに基づいて求める。続いて、ステップＳ１４１で、ページフォールトが発生したページをＲＥＡＤ可能に設定する。

【００３８】

次に図１５を参照して、ＷＲＩＴＥページフォールトが発生が発生した場合のクラスタ共有メモリ用ページ操作手段１１５処理手順を説明する。まず、ステップＳ１５０において、ページフォールトが発生したページがＲＥＡＤ可能に設定されているかどうかを調べる。ＲＥＡＤ可能な場合には、ステップＳ１５２へ進む。ＲＥＡＤ可能でない場合には、ステップＳ１５１に進む。

【００３９】

ステップＳ１５１では、ページフォールトが発生した共有メモリ内のページのデータをクラスタ共有ファイルシステム１１７によりクラスタ共有ファイル（共有ディスク装置１６）の対応する部分から当該共有メモリ内のページに読み込む。この時、ページのデータをクラスタ共有ファイルのどの部分から読み込むかは

、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８に登録されたデータに基づいて求める。

【００４０】

続いて、ステップＳ１５２で、ページフォールトが発生した共有メモリ内のページをＲＥＡＤ／ＷＲＩＴＥ可能に設定する。最後に、ステップＳ１５３で、ページフォールトが発生した共有メモリ内のページの番号を図６に示す更新ページリスト１１９に登録する。クラスタ共有メモリ用ページ操作手段１１５は、このようにページフォールト処理をした後、実際にクラスタ共有メモリへアクセスする。

【００４１】

このようにステップＳ７２で共有メモリへのアクセスをした後、続くステップＳ７３において、プロセス１１２は、クラスタ共有メモリ用ロックを解放するためにクラスタ共有メモリ用ロック操作手段１１４に指示して、クラスタ共有メモリロックのアンロック操作を実行させる。

【００４２】

ここで図１３を参照して、クラスタ共有メモリ用ロック操作手段１１４によるクラスタ共有メモリロックのアンロック操作手順を説明する。図１３は、クラスタ共有メモリ用ロック操作手段１１４によるクラスタ共有メモリロックのアンロック操作の操作手順を示すフローチャート図である。

【００４３】

まず、ステップＳ１３０において、図６に示した更新ページリスト１１９に登録されている全ての更新ページのデータをクラスタ共有ファイルシステム１１７により共有ディスク装置１６に記録されているクラスタ共有ファイルの該当部分に書き込む。この時、更新ページをクラスタ共有ファイルのどの部分に書き込むかは、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８を用いて求める。　続いて、ステップＳ１３１において、更新ページリスト１１９をクリアする。

【００４４】

以上で図７に示したクラスタ共有メモリを用いたクラスタ共有ファイルへのア

クセス動作の処理手順の説明を終了する。ここで、プロセス１１２は、これまでの説明でアクセスした共有ファイルをまだアクセスする場合には、図４に示したクラスタ共有メモリ用ロック／クラスタ共有ファイル用ロック変換テーブル１１３のエントリーを削除せずに残しておく。このエントリーを残しておけば、次にロックを取得する場合には、クラスタ共有メモリ用ロック操作手段１１４がクラスタ共有ファイルシステム用ロック機構１１６にファイルのロック取得を依頼するだけでロック取得ができる。

【００４５】

もし、プロセス１１２がこれまでの説明でアクセスした共有ファイルを以後アクセスすることがない場合には、プロセス１１２は、クラスタ共有メモリ用ロック解放手段１２３に指示してクラスタ共有メモリ用ロックを解放させる。図１１は、クラスタ共有メモリ用ロック解放手段１２３によるクラスタ共有メモリ用ロックを解放させる処理手順を示すフローチャート図である。図１１において、ステップＳ１１０で、プロセス１１２が指定したクラスタ共有メモリ用ロックに関するエントリを図４に示すクラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル１１３から抹消する。
これにより、クラスタ共有メモリ用ロックが解放される。

【００４６】

最後にプロセス１１２が共有メモリへのアクセスとして処理すべき全ての共有ファイルへのアクセスが終了した場合には、プロセス１１２がクラスタ共有メモリアンマップ手段１２１に指示してプロセスのアドレス空間内に仮想的に設定した共有メモリのマッピングを解除させる。すなわち、クラスタ共有メモリをアンマップさせる。

【００４７】

図９は、クラスタ共有メモリアンマップ手段１２１がクラスタ共有メモリをアンマップする処理手順を示すフローチャート図である。図９において、まず、ステップＳ９０において、プロセス１１２のアドレス空間内にアロケーションしたクラスタ共有メモリのための領域を解放する。続いて、ステップＳ９１において、アンマップする領域に関するエントリを図５に示すクラスタ共有メモリ/クラ

スタ共有ファイル変換テーブル１１８から抹消する。

【００４８】

次に図１６を用いて本発明において、２つのサーバーコンピュータが共有ディスク装置に記録されている同じファイルに対するアクセスをする場合の動作を説明する。図１６は、サーバーコンピュータＡとサーバーコンピュータＢと共有ディスク装置Ｃとから構成されるクラスタシステムの動作を説明するためのシステム構成を示す図である。。

【００４９】

図１６において、サーバーコンピュータＡとサーバーコンピュータＢとは、されざれ共有ディスク装置Ｃに接続されている。サーバーコンピュータＡでは、プロセスＡが動作している。また、サーバーコンピュータＢでは、プロセスＢが動作している。これらのプロセスＡとプロセスＢは、それぞれ図示を省略したクラスタ共有ファイルシステムで管理されている共有ディスク装置Ｃに記録されたクラスタ共有ファイルＤを先に説明したように自身のアドレス空間内にマッピングしているものとする。クラスタ共有ファイルＤの中にはデータ領域Ｘが存在する。

【００５０】

このように設定された状況の下に、プロセスＡが（１）～（３）の処理を実行し、次にプロセスＢが（４）～（６）の処理を実行し、更に再びプロセスＡ（７）～（９）の処理を実行するとする。まずプロセスＡは（１）でクラスタ共有メモリ用ロックを取得する。これによりクラスタ共有ファイルＤがクラスタ共有メモリとしてマップされた領域のページは全てアクセス不許可になる。次に（２）でクラスタ共有メモリ上のデータ領域Ｘに数値「１」を加えようとする。するとＷＲＩＴＥページフォールトが発生する。クラスタ共有メモリ操作手段は、この領域がＲＥＡＤ不可能なので、このデータ領域を含むページの内容をクラスタ共有ファイルＤから読みこむ。

【００５１】

そして、そのページをＲＥＡＤ／ＷＲＩＴＥ可能に設定し、ページフォールト処理を終了する。ページフォールトから戻った後、プロセスＡはデータ領域Ｘの

値に１を加える。そして（３）でクラスタ共有メモリ用ロックを解放する。これにより、クラスタ共有メモリ用ロック操作手段は、更新ページであるデータ領域Ｘを含むページの内容をクラスタ共有ファイルＤに書き戻す。データ領域Xの初期値が「０」だったとすると、この時点でデータ領域Ｘの値は「１」となる。

【００５２】

次にプロセスＢは（４）でクラスタ共有メモリ用ロックを取得する。これによりクラスタ共有ファイルＤ4がクラスタ共有メモリとしてマップされた領域のページは全てアクセス不許可になる。次に（５）でクラスタ共有メモリ上のデータ領域Ｘに数値「１」を加えようとする。するとＷＲＩＴＥページフォールトが発生する。クラスタ共有メモリ操作手段は、この領域がＲＥＡＤが不可能なので、このデータ領域を含むページの内容をクラスタ共有ファイルＤから読みこむ。そして、そのページをＲＥＡＤ／ＷＲＩＴＥ可能に設定し、ページフォールト処理を終了する。ページフォールトから戻った後、プロセスＢはデータ領域Ｘの値に数値「１」を加える。そして（６）でクラスタ共有メモリ用ロックを解放する。これにより、クラスタ共有メモリ用ロック操作手段は、更新ページであるデータ領域Ｘを含むページの内容をクラスタ共有ファイルＤに書き戻す。この時点でデータ領域Ｘの値は「２」になる。

【００５３】

最後にプロセスＡは（７）でクラスタ共有メモリ用ロックを取得する。これによりクラスタ共有ファイルＤがクラスタ共有メモリとしてマップされた領域のページは全てアクセス不許可になる。次に（８）でクラスタ共有メモリ上のデータ領域Xに１を加えようとする。するとＷＲＩＴＥページフォールトが発生する。クラスタ共有メモリ操作手段は、この領域がＲＥＡＤ／ＷＲＩＴＥ不可能なので、このデータ領域を含むページの内容をクラスタ共有ファイルＤから読みこむ。そして、そのページをすＲＥＡＤ／ＷＲＩＴＥ可能に設定し、ページフォールト処理を終了する。ページフォールトから戻った後、プロセスＡはデータ領域Ｘの値に数値「１」を加える。そして（９）でクラスタ共有メモリ用ロックを解放する。これにより、クラスタ共有メモリ用ロック操作手段は、更新ページであるデ

ータ領域Ｘを含むページの内容をクラスタ共有ファイルＤに書き戻す。この時点でデータ領域Ｘの値は３になる。

【００５４】

【発明の効果】

本発明を適用することにより、クラスタ共有ファイルシステムを持つクラスタシステムで、ファイルへの共有アクセスと共に、メモリへの共有アクセスも可能になる。

更に本発明では、クラスタ共有ファイルシステムを用いることで、クラスタ共有メモリを安価に実現することができる。また、クラスタ共有ファイルをクラスタ共有メモリとしてマッピングするので、更新データに永続性が持たれる。更に、本発明では、クラスタ共有メモリを、クラスタ共有ファイルシステムを用いて実現するため、クラスタ共有ファイルを分散共有メモリとしてマッピングし、メモリとしてのアクセス（ｌｏａｄ命令／ｓｔｏｒｅ命令）とファイルとしてのアクセス（ｒｅａｄシステムコール／ｗｒｉｔｅシステムコール）を並列に実行することができるようになる。

【図面の簡単な説明】

【図１】

本発明のクラスタシステム１０を示す図である。

【図２】

サーバーコンピュータ１１の構成を示す図である。

【図３】

プロセス１１２の詳細を示した図でする。

【図４】

クラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル１１３の詳細を示した図である。

【図５】

クラスタ共有メモリ/クラスタ共有ファイル変換テーブル１１８の詳細を示した図である。

【図６】

更新ページリスト１１９の詳細を示した図である。

【図７】

クラスタ共有メモリを用いたクラスタ共有ファイルへのアクセス動作の処理手順を示すフローチャート図である。

【図８】

クラスタ共有メモリマップ手段１２０による共有メモリのマッピング操作手順を示すフローチャート図である。

【図９】

クラスタ共有メモリアンマップ手段１２１がクラスタ共有メモリをアンマップする処理手順を示すフローチャート図である。

【図１０】

クラスタ共有メモリ用ロックアロケーション手段１２２によるクラスタ共有メモリ用ロックアロケーション操作手順を示すフローチャート図である。

【図１１】

クラスタ共有メモリ用ロック解放手段１２３によるクラスタ共有メモリ用ロックを解放させる処理手順を示すフローチャート図である。

【図１２】

クラスタ共有メモリ用ロック操作手段１１４のロック操作の処理手順を示すフローチャート図である。

【図１３】

クラスタ共有メモリ用ロック操作手段１１４によるクラスタ共有メモリロックのアンロック操作の操作手順を示すフローチャート図である。

【図１４】

ＲＥＡＤページフォールトが発生が発生した場合のクラスタ共有メモリ用ページ操作手段１１５の処理手順を示すフローチャート図である。

【図１５】

ＷＲＩＴＥページフォールトが発生が発生した場合のクラスタ共有メモリ用ページ操作手段１１５の処理手順を示すフローチャート図である。

【図１６】

２つのサーバーコンピュータが共有ディスク装置に記録されている同じファイルに対するアクセスをする場合の動作を説明するためのシステム構成を示す図である。

【符号の説明】

１１・・・・サーバーコンピュータ

１２・・・・サーバーコンピュータ

１３・・・・サーバーコンピュータ

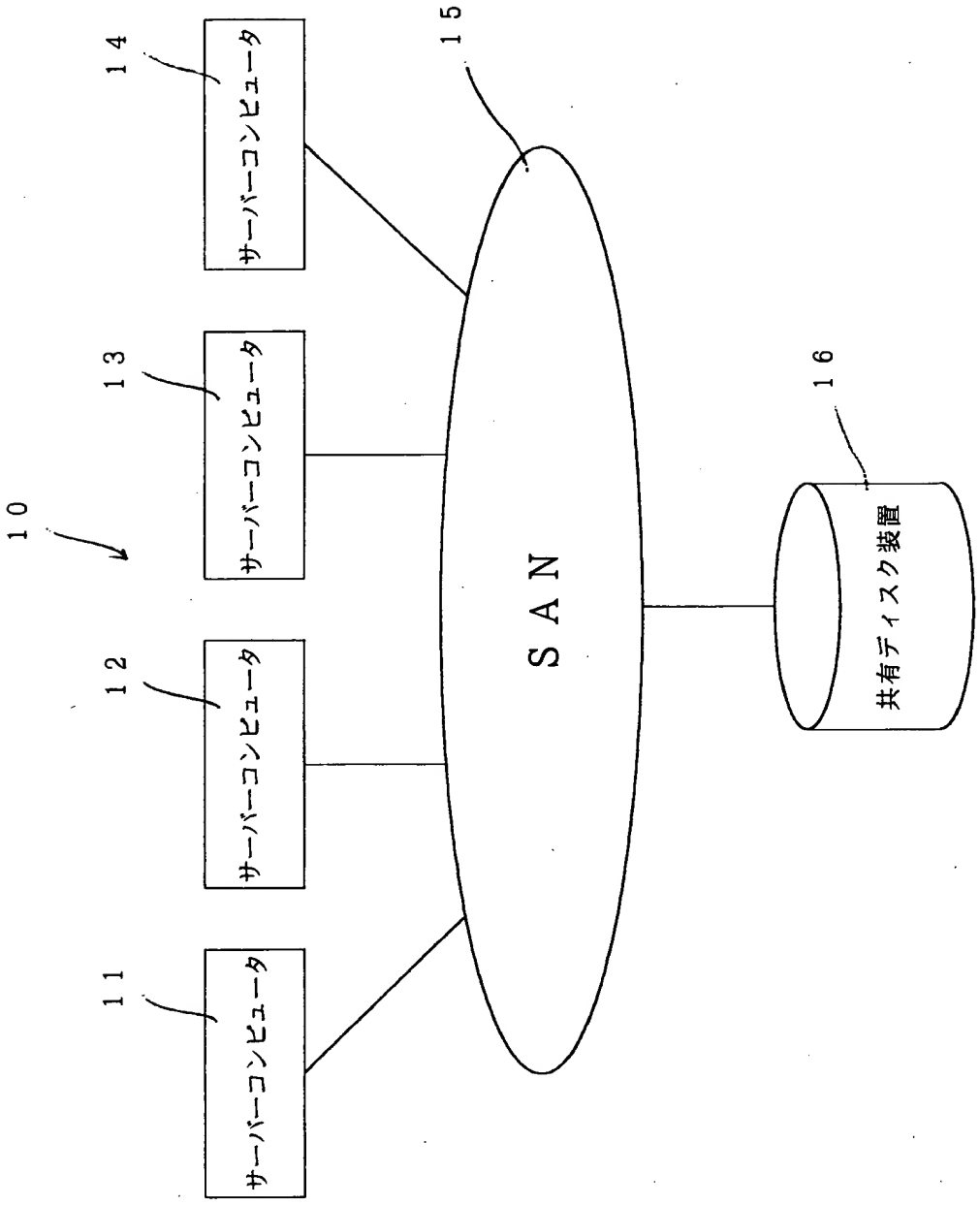
１４・・・・サーバーコンピュータ

１５・・・・ＳＡＮ

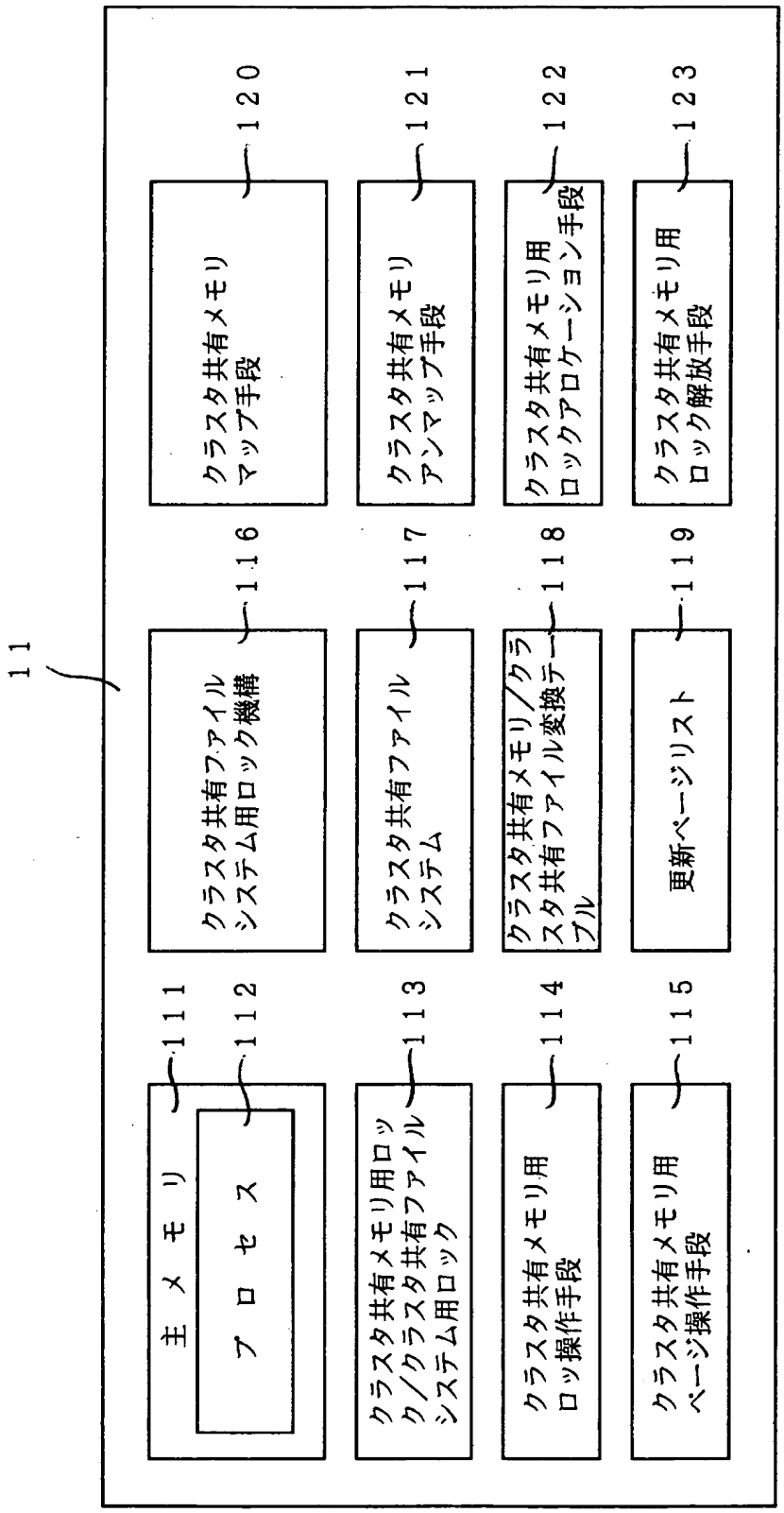
１６・・・・共有ディスク装置

【書類名】　図面

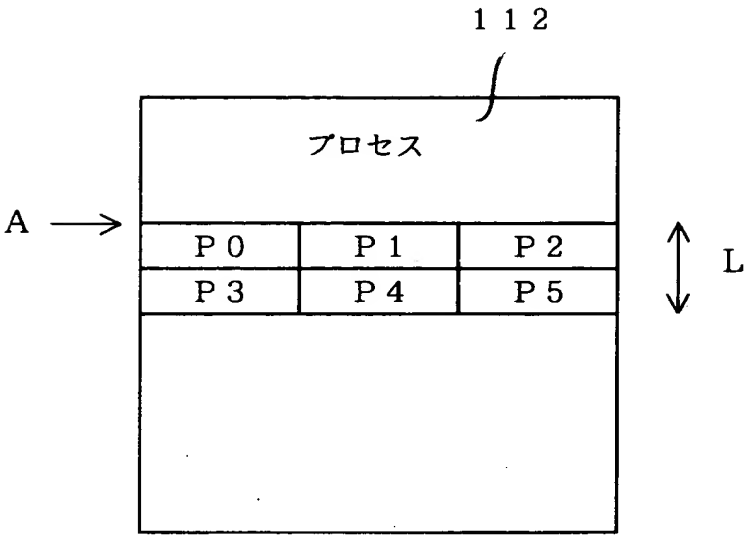
【図１】


１０
１１
サーバーコンピュータ
１２
サーバーコンピュータ
１３
サーバーコンピュータ
１４
サーバーコンピュータ
１５
ＳＡＮ
１６
共有ディスク装置

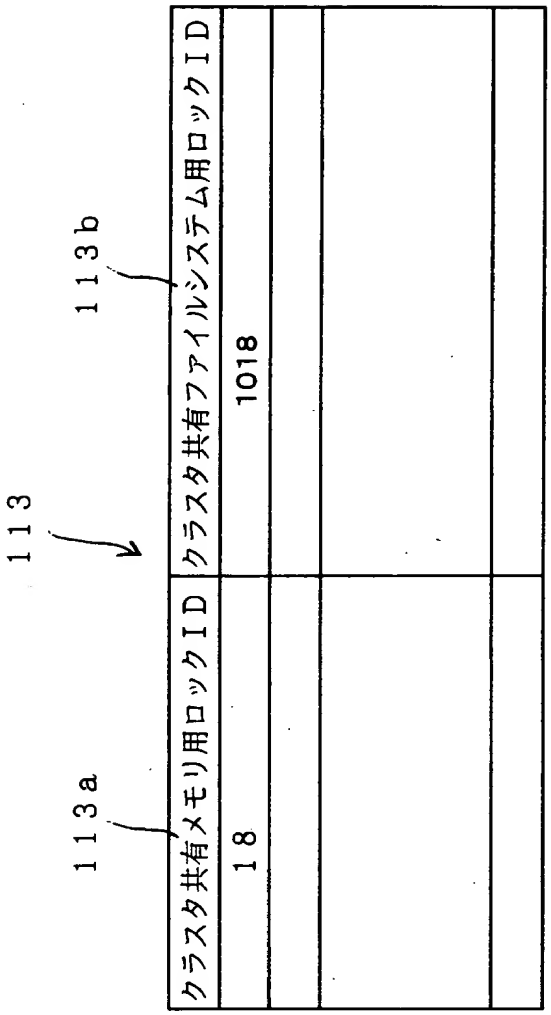
【図２】


11
主メモリ
111
プロセス
112
クラスタ共有メモリ用ロック／クラスタ共有ファイルシステム用ロック
113
クラスタ共有メモリ用ロック操作手段
114
クラスタ共有メモリ用ページ操作手段
115
クラスタ共有ファイルシステム用ロック機構
116
クラスタ共有ファイルシステム
117
クラスタ共有メモリ／クラスタ共有ファイル変換テーブル
118
更新ページリスト
119
クラスタ共有メモリマップ手段
120
クラスタ共有メモリアンマップ手段
121
クラスタ共有メモリ用ロックアクケーション手段
122
クラスタ共有メモリ用ロック解放手段
123

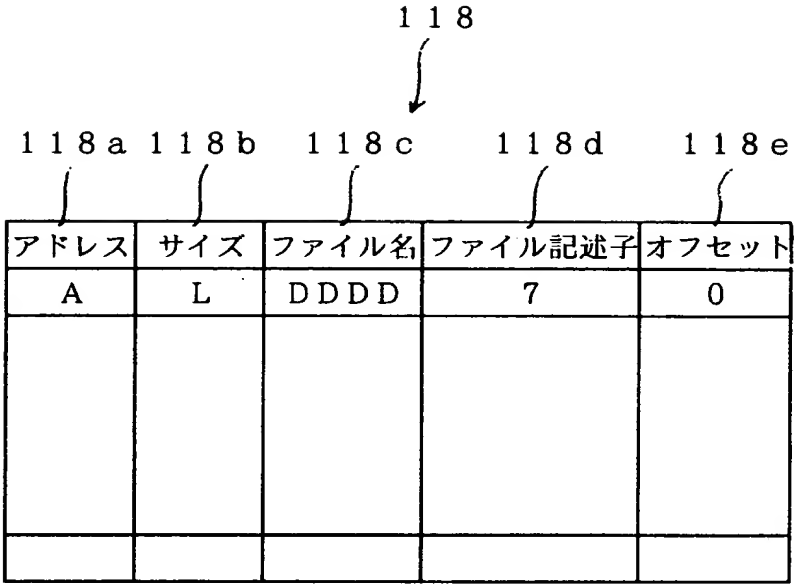
【図３】



【図４】

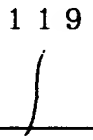

113
113a
113b
クラスタ共有メモリ用ロックＩＤ
クラスタ共有ファイルシステム用ロックＩＤ
18
1018

【図５】



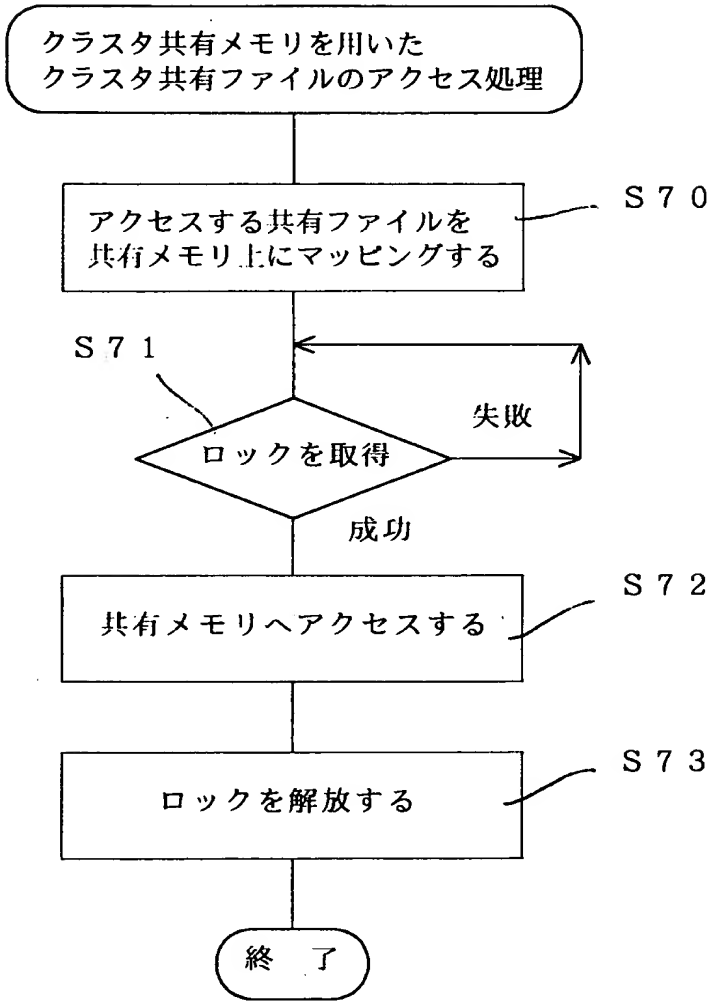
アドレス	サイズ	ファイル名	ファイル記述子	オフセット
A	L	DDDD	7	0

【図６】

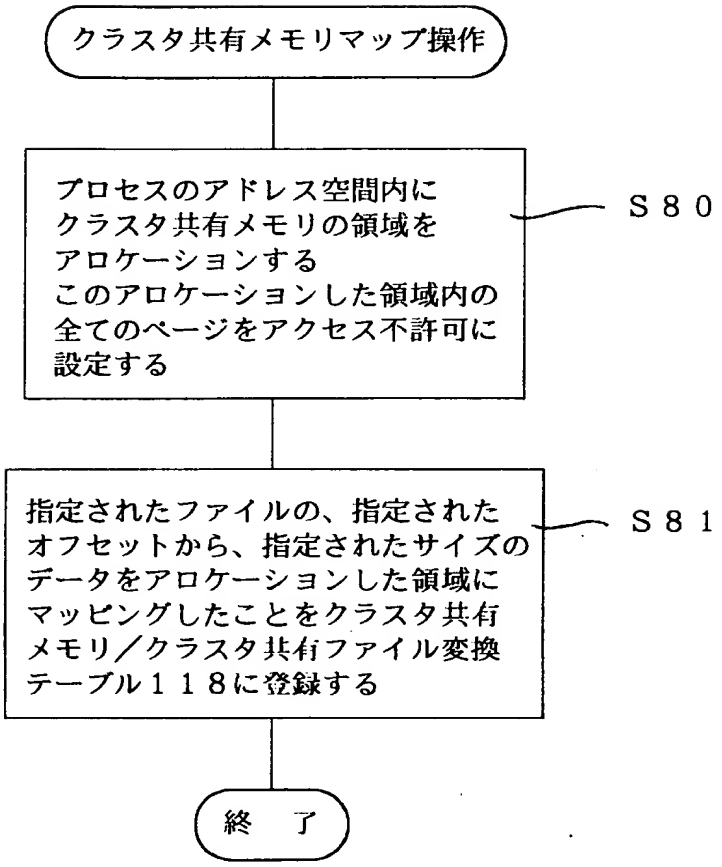


P４	P２		

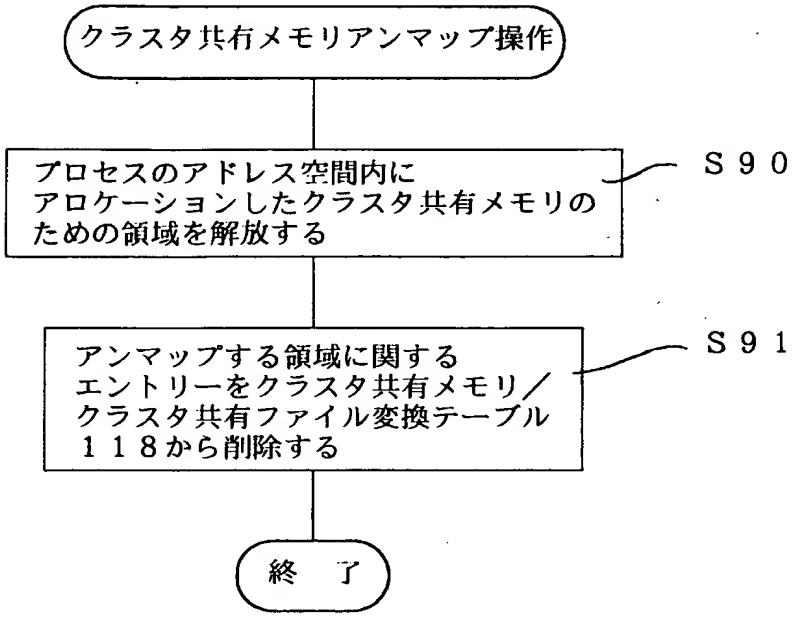
【図７】


クラスタ共有メモリを用いた
クラスタ共有ファイルのアクセス処理
アクセスする共有ファイルを
共有メモリ上にマッピングする
Ｓ７０
Ｓ７１
ロックを取得
失敗
成功
共有メモリへアクセスする
Ｓ７２
ロックを解放する
Ｓ７３
終　了

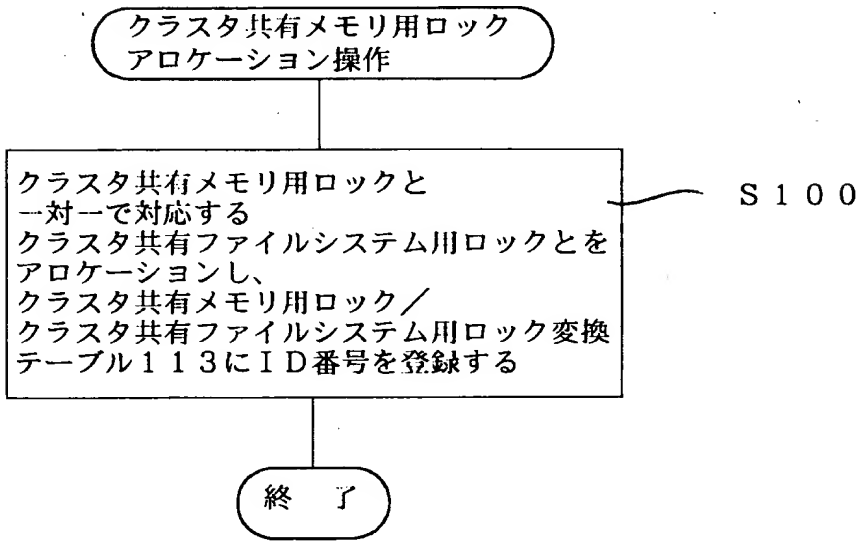
【図８】


クラスタ共有メモリマップ操作
プロセスのアドレス空間内に
クラスタ共有メモリの領域を
アロケーションする
このアロケーションした領域内の
全てのページをアクセス不許可に
設定する
Ｓ８０
指定されたファイルの、指定された
オフセットから、指定されたサイズの
データをアロケーションした領域に
マッピングしたことをクラスタ共有
メモリ／クラスタ共有ファイル変換
テーブル１１８に登録する
Ｓ８１
終　了

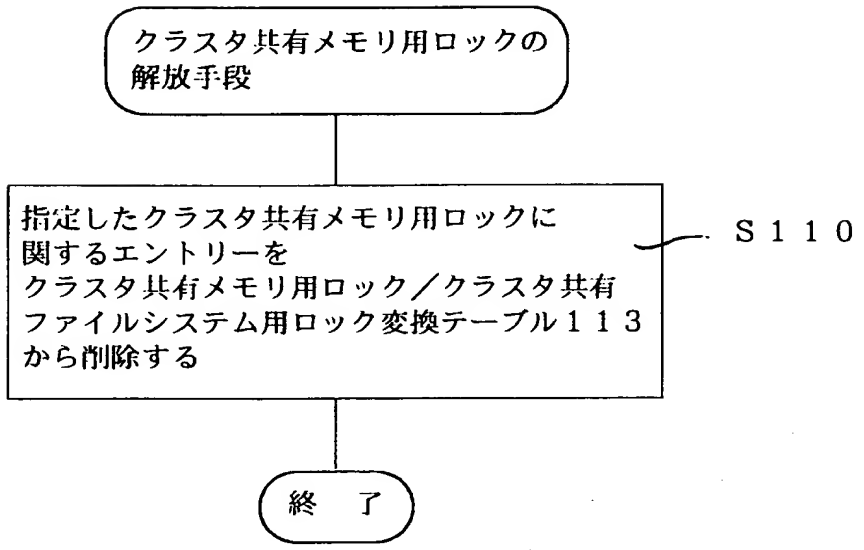
【図９】


クラスタ共有メモリアンマップ操作
プロセスのアドレス空間内に
アロケーションしたクラスタ共有メモリの
ための領域を解放する
Ｓ９０
アンマップする領域に関する
エントリーをクラスタ共有メモリ／
クラスタ共有ファイル変換テーブル
１１８から削除する
Ｓ９１
終　了

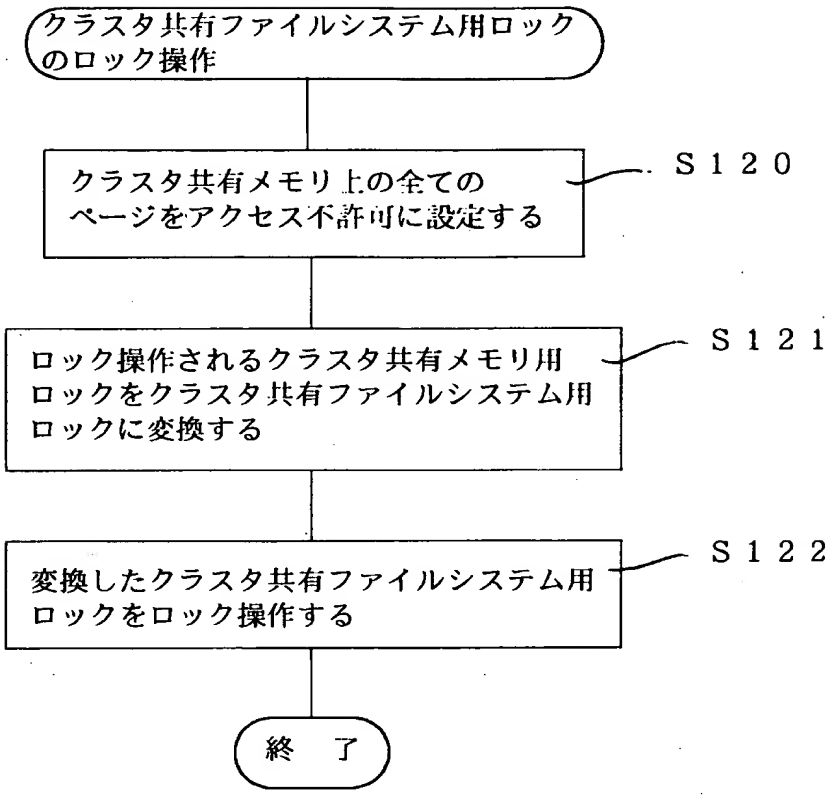
【図１０】


クラスタ共有メモリ用ロック
アロケーション操作
クラスタ共有メモリ用ロックと
一対一で対応する
クラスタ共有ファイルシステム用ロックとを
アロケーションし、
クラスタ共有メモリ用ロック／
クラスタ共有ファイルシステム用ロック変換
テーブル１１３にＩＤ番号を登録する
Ｓ１００
終　了

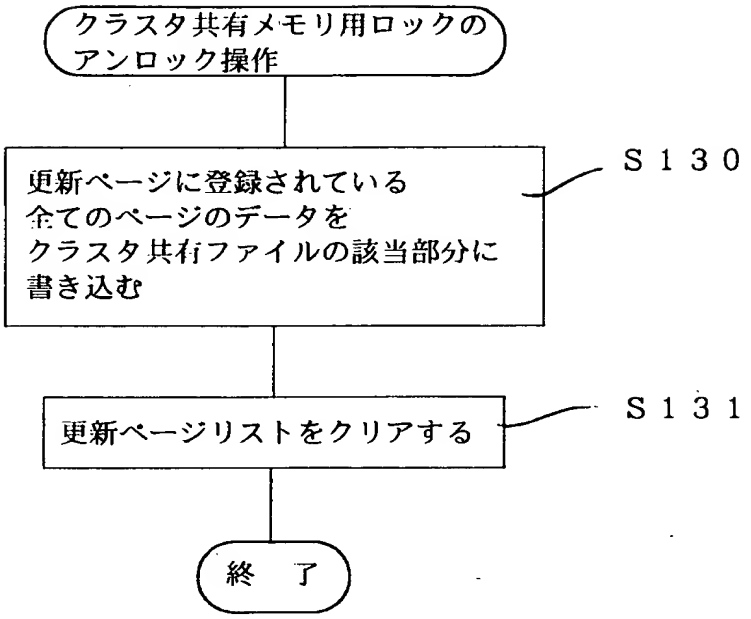
【図１１】


クラスタ共有メモリ用ロックの
解放手段
指定したクラスタ共有メモリ用ロックに
関するエントリーを
クラスタ共有メモリ用ロック／クラスタ共有
ファイルシステム用ロック変換テーブル１１３
から削除する
Ｓ１１０
終　了

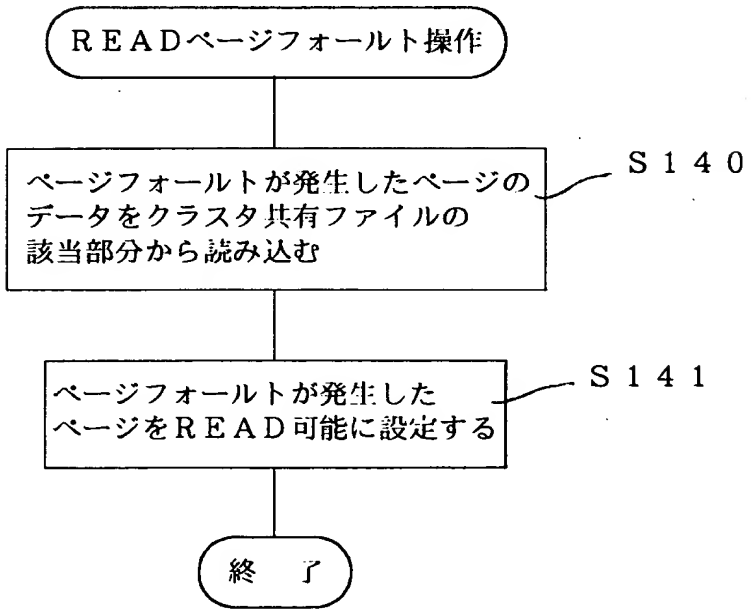
【図１２】


クラスタ共有ファイルシステム用ロックのロック操作
クラスタ共有メモリ上の全てのページをアクセス不許可に設定する
Ｓ１２０
ロック操作されるクラスタ共有メモリ用ロックをクラスタ共有ファイルシステム用ロックに変換する
Ｓ１２１
変換したクラスタ共有ファイルシステム用ロックをロック操作する
Ｓ１２２
終　了

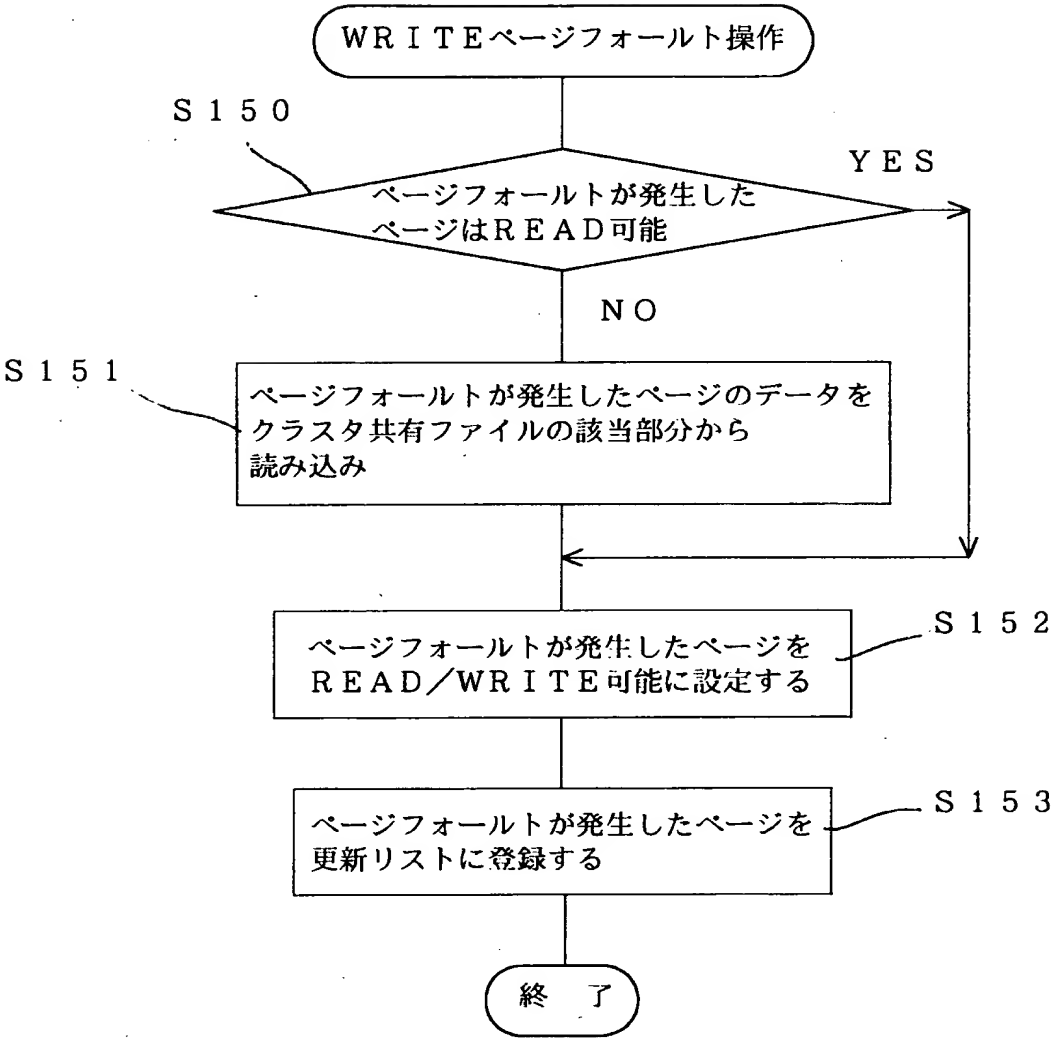
【図１３】


クラスタ共有メモリ用ロックの
アンロック操作
更新ページに登録されている
全てのページのデータを
クラスタ共有ファイルの該当部分に
書き込む
Ｓ１３０
更新ページリストをクリアする
Ｓ１３１
終　了

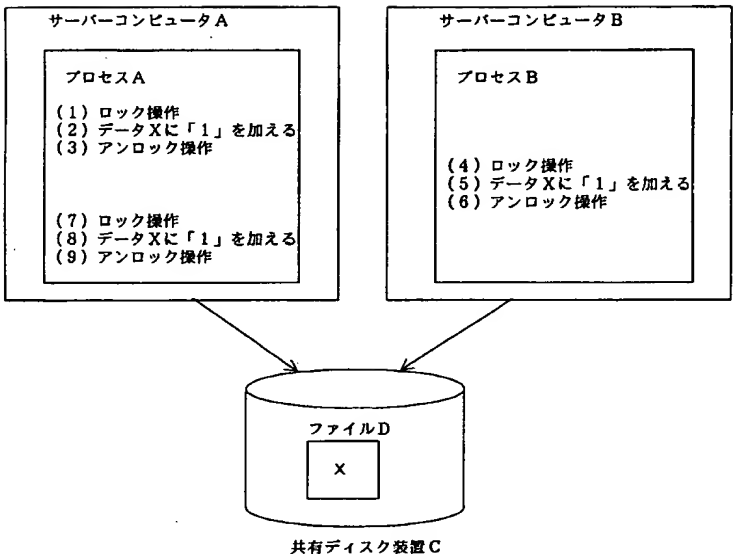
【図１４】


ＲＥＡＤページフォールト操作
ページフォールトが発生したページの
データをクラスタ共有ファイルの
該当部分から読み込む
Ｓ１４０
ページフォールトが発生した
ページをＲＥＡＤ可能に設定する
Ｓ１４１
終　了

【図１５】


ＷＲＩＴＥページフォールト操作
Ｓ１５０
ページフォールトが発生した
ページはＲＥＡＤ可能
ＹＥＳ
ＮＯ
Ｓ１５１
ページフォールトが発生したページのデータを
クラスタ共有ファイルの該当部分から
読み込み
ページフォールトが発生したページを
ＲＥＡＤ／ＷＲＩＴＥ可能に設定する
Ｓ１５２
ページフォールトが発生したページを
更新リストに登録する
Ｓ１５３
終　了

【図１６】


サーバーコンピュータＡ
プロセスＡ
（１）ロック操作
（２）データＸに「１」を加える
（３）アンロック操作
（７）ロック操作
（８）データＸに「１」を加える
（９）アンロック操作
サーバーコンピュータＢ
プロセスＢ
（４）ロック操作
（５）データＸに「１」を加える
（６）アンロック操作
ファイルＤ
Ｘ
共有ディスク装置Ｃ

【書類名】　要約書

【要約】

【課題】　本発明は、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムにおいて、並列プログラムの記述が容易なクラスタシステムを提供することを目的とする。

【解決手段】　クラスタシステムにおいて、アプリケーションプログラムを実行することで生成されるプロセスが共有ディスク装置に記録されたファイルをクラスタ共有ファイルシステムを用いてプロセスが主メモリ上に配置されているアドレス空間内に仮想的に設けたクラスタ共有メモリ（分散共有メモリ）領域上にマッピングすることで、ファイルへのアクセスを主メモリへのアクセスとして処理する。

【選択図】　図１

認定・付加情報

特許出願の番号	特願２００１－０７１６８０
受付番号	５０１００３６０７０１
書類名	特許願
担当官	第七担当上席　００９６
作成日	平成１３年　３月１５日

＜認定情報・付加情報＞

【提出日】　平成13年 3月14日

次頁無

出　願　人　履　歴　情　報

識別番号　　〔０００００３０７８〕

１．変更年月日　　１９９０年　８月２２日
　［変更理由］　新規登録
　　住　所　　神奈川県川崎市幸区堀川町７２番地
　　氏　名　　株式会社東芝

２．変更年月日　　２００１年　７月　２日
　［変更理由］　住所変更
　　住　所　　東京都港区芝浦一丁目１番１号
　　氏　名　　株式会社東芝